REMARKS

Claims 1-29 are now pending in this application. The Office action mailed on October 3, 2003 ("Office action") rejected claims 1-3, 6-10, 13-25, 27, and 29 under 35 U.S.C. §102(a) as being anticipated by U.S. Patent No. 5,956,023 to Lyle et al. ("Lyle"). Additionally, claims 4-5 and 26 were also rejected under 35 U.S.C. §102(a) (because the Office action has argued so under the same heading). Furthermore, claim 28 was rejected under 35 U.S.C. §103(a) as being unpatentable over Lyle in view of U.S. Patent No. 6,418,421 to Hurtado et al. ("Hurtado") and claims 11-12 were rejected under 35 U.S.C. §103(a) as being unpatentable over Lyle in view of U.S. Patent No. 6,515,656 to Wittenberg et al. ("Wittenberg"). Applicants respectfully, but strongly, disagree.

By present amendment, claim 1 has been amended for clarification and not in view of the prior art and/or for purposes related to patentability. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims, and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on January 7, 2004. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a method and system for interacting with a modeling layout algorithm in a computing environment having two engines. An "engine" is a term widely recognized in the art and a generally accepted definition, as defined by the Microsoft Press Computer Dictionary is as follows: a back-end processor or portion of a program that determines how the program manages and manipulates data. The term "engine" is most often used in relation to a specific program; for example, a database engine contains the tools for manipulating a database.

Thus, the method and system of the present invention is directed to enabling incremental and interruptible automatic layout operations (operations that are very time-consuming and resource-intensive) that are conducted between two specific engines called a layout engine and a modeling engine. Each of these engines work in conjunction to provide an automatic layout process for the modeling of specific designs, such as a microchip layout or a motherboard layout.

Embodiments of the present invention further comprise a defined set of interfaces (*e.g.,* of a COM object) between a layout engine and a modeling engine. In one aspect, the modeling engine calls upon these interfaces to start and stop the layout process, preserve and restore state information, and perform other functions. In another aspect, the layout engine, which may be a pluggable component in the computing environment, raises events through another interface

to indicate when the layout engine may be safely interrupted or to indicate progress. Then, the modeling engine can call back into the layout engine to stop the layout if the user has requested that the layout be interrupted. As such, using interfaces between the modeling engine and various layout engines, one may interrupt a layout process while preserving and eventually restoring its state, thereby enabling incremental layout operations that do not lose progress. Note that the above description is for example and informational purpose only and should not be used to interpret the claims, which are discussed below.

Turning to the claims, claim 1 recites a system in a computing device comprising a modeling engine, . . . a layout engine, . . . and a set of at least one interface connecting the modeling engine to the layout engine . . .. The Office action contends that Lyle teaches each of the limitations recited in claim 1. Specifically, the Office action contends that Lyle teaches a modeling engine at column 4, lines 54-64 by disclosing an application control manager. Additionally, the Office action contends that Lyle teaches a layout engine at column 10, lines 3-14 by disclosing the transmitting of a prescribed function code to the application control manager to automatically initiate a process. Finally, the Office action contends that Lyle teaches a set of at least one interface connecting the modeling engine to the layout engine at column 10, lines 30-40 by disclosing a pause/end pushbutton for user interaction with the automatic layout process. Applicants respectfully disagree with each of these premises.

In a first element, claim 1 recites a modeling engine for editing modeling elements connected to a user interface. Engine (sometimes called a back-end

processor) is a term widely used in the industry to describe a piece of hardware that encapsulates some function but that cannot be used without some kind of controller, often called a front-end processor. Thus, a modeling engine for editing modeling elements, as recited in claim 1, is a particular kind of engine that is designed to perform resource-intensive tasks such as emulating (modeling) microelectronic system designs where model elements behave and interact with each as though they were real as defined by the parameters stored in the modeling engine. More specifically, as stated in the specification of the present invention at page 16, second paragraph, a modeling engine provides additional support for rendering and editing models beyond that available via a less application-specific rendering engine in a computer system for rendering models. In other words, a modeling engine provides additional capabilities that a typical rendering engine does not natively provide.

For example, four-point hit testing, which is one way to determine whether one model element is above or in close proximity to another model element in a modeling environment, is not natively handled by a typical rendering engine, but is instead provided by a modeling engine. Moreover, in one aspect of the present invention, the modeling engine includes layout-related methods to handle the interface with a layout engine component. See pages 16-17 of the specification of the present invention. As such, the term modeling engine is a recognized term of art in the industry and carries specific meaning as recognized by those skilled in this art and its use in the present invention does not deviate from the industry-specific meaning.

In contrast, the cited and applied portion (see column 4, lines 54-64) of Lyle teaches an application control manager which the Office action contends is a modeling engine. An application control manager is a controller or a front-end processor. Specifically, Lyle discloses that the application control manager administers activities of a library of control applications wherein each control application includes a set of instructions for controlling a piece of hardware, such as a pump. That is, the application control manager in Lyle is a front-end processor that may control other processes, but is not intended to handle resource-intensive tasks that a typical engine (back-end processor) is designed for. Simply stated, a front-end processor, *i.e.*, an application control manager, cannot be inferred to be an engine, *i.e.*, a back-end processor, let alone a highly specialized kind of engine such as a modeling engine for editing modeling elements as recited in claim 1.

Turning to another element, claim 1 recites a layout engine connected to the modeling engine and configured to execute an automatic layout process that automatically lays out modeling elements. Again, engine (a back-end processor) is a term widely used in the industry to describe a piece of hardware that encapsulates some function but can't be used without some kind of front-end processor. Thus, the layout engine of claim 1 is a highly specialized type of engine configured to execute an automatic layout process that automatically lays out modeling elements.

In contrast, the cited and applied portion (see column 10, lines 3-14) of Lyle teaches touch selectable button fields that, when activated, cause an interface

manager to transmit a prescribed function code to the application control manager (a front-end processor). The button fields are merely input points to the system and are operable to send a control signal to the application control manager. This arrangement cannot be inferred to be an engine configured to execute a process, let alone a layout engine configured to execute an automatic layout process that automatically lays out modeling elements. Lyle does not teach modeling elements, an automatic layout process, or even an engine.

Further, claim 1 recites a set of at least one interface connecting the modeling engine to the layout engine, the set including at least one interface through which the modeling engine communicates with the layout engine to provide user interaction with the automatic layout process other than to cancel the automatic layout process. That is, the system of claim 1 includes an interface that allows communication between two engines during an automatic layout process and by virtue of that communication, the automatic layout process may be paused or saved, *i.e.*, not simply cancelled.

The cited and applied portion (column 10, lines 30-40) of Lyle disclose a variety of button fields that include a pause/end button that is capable of pausing a process implemented by the application control manager. However, the processes disclosed in Lyle are not conducted by any engine, let alone two highly-specialized types of engines such as a layout engine and a modeling engine. Lyle simply discloses the pausing of any process which is far simpler than pausing a back-end processor engaged in its resource-intensive, highly-specialized task. By analogy, the pause button on a CD player is another example of what is disclosed by Lyle.

As stated above, Lyle does not teach a modeling engine, a layout engine, or an automatic layout process. Therefore, Lyle cannot possibly teach an interface which a modeling engine communicates with a layout engine to provide a user control over an automatic layout process.

For at least these reasons, applicants submit that claim 1 is patentable over the prior art of record.

Claims 2-13 depend either directly or indirectly from claim 1. Applicants further submit that claims 2-13 are also allowable for the additional patentable elements included in these claims.

For example, claim 2 recites that the modeling engine communicates with the layout engine by calls from the layout engine via the interface. As shown above, Lyle does not teach any engine, let alone a highly-specialized modeling engine and a layout engine. Nor does Lyle teach communication between a modeling engine and a layout engine initiated by a call from the layout engine. Applicants submit that claim 2 is allowable for at least this additional reason.

In another example, claim 6 recites that the modeling engine communicates with the layout engine to interrupt the automatic layout process. Again, Lyle does not teach any engine, let alone a highly-specialized modeling engine and a layout engine, or communication between a modeling engine and a layout engine wherein an automatic layout process is interrupted. Applicants submit that claim 6 is allowable for at least this additional reason.

In yet another example, with regard to claims 11 and 12, the Office action contends that Lyle fails to teach a system wherein the layout engine comprises a

pluggable software component. The Office action goes on further to state that Wittenberg teaches usage and implementation of pluggable software and states that it would have been obvious to an artisan at the time of the invention to include Wittenberg's teachings with Lyle's teachings in order to allow a user to add components that are independent of other components. This is contended for both claims 11 and 12. Applicants respectfully disagree with this conclusion.

Applicants acknowledge that Wittenberg may teach a pluggable software component. However, Lyle again does not teach virtually any element of claim 1, the claim from which claims 11 and 12 depend. That is, as shown above, Lyle does not teach a modeling engine, a layout engine, or an automatic layout process. Thus, the combination of Lyle and Wittenberg, even if somehow permissible, would not make obvious the recitations of claims 11 and 12 because the recited limitations are not met by any such combination. Applicants submit that claims 11 and 12 are allowable for at least this additional reason.

Turning to independent clam 14, claim 14 recites a computer-implemented method, comprising, starting a layout engine to lay out model elements, receiving information from the layout engine indicating that it can be safely interrupted, and interrupting the layout engine based on the information. The Office action contends that Lyle teaches each of the limitations recited in claim 14. Specifically, the Office action contends that Lyle teaches starting a layout engine to lay out model elements at FIG. 5, item 98 by disclosing a picture of a system initialization screen shot. Additionally, the Office action contends that Lyle teaches receiving information from the layout engine indicating that it can be safely interrupted, and

interrupting the layout engine based on the information at FIG. 9, item 86, by disclosing a touch-screen pushbutton. Applicants respectfully disagree with each of these contentions.

The cited and applied portion of Lyle again does not teach or event suggest using an engine, let alone using a highly-specialized type of engine called a layout engine that is capable of laying out model elements. Further, the touch-screen pushbutton shown on FIG. 9 is merely a user input interface. That is, the pushbutton is not capable of indicating that a process can be safely interrupted. When activated, the system is simply interrupted without any regard to information received indicating a safe environment for interruption or otherwise. For at least these reasons, applicants submit that claim 14 is patentable over the prior art of record.

With regard to claims 15-18, these claims depend either directly or indirectly from claim 14. Applicants further submit that claims 15-18 are also allowable for the additional patentable elements included in these claims.

For example, claim 16 recites that the method of claim 15 further comprises receiving a request to interrupt the layout engine, and waiting for the information from the layout engine indicating that it can be safely interrupted. As shown above, Lyle does not teach any engine, let alone a highly-specialized layout engine, and also does not teach communication to a layout engine to interrupt an automatic layout process only after receiving permission. Lyle merely describes a pushbutton that, once activated, immediately interrupts a process without regard to waiting for

permission as recited in claim 16. Applicants submit that claim 16 is allowable for at least this additional reason.

Finally, turning to independent claim 19, claim 19 recites a computer-implemented method, comprising starting a layout engine to lay out model elements, providing information to the layout engine by which the layout engine preserves state information, interrupting the layout engine, providing information to the layout engine by which the layout engine restores state from the state information, and restarting the layout engine from the restored state. The Office action contends that Lyle teaches each of the limitations recited in claim 19. Specifically, the Office action contends that Lyle teaches starting a layout engine to lay out model elements at column 10, lines 3-14 by disclosing the transmitting of a prescribed function code to the application control manager to automatically initiate a process. Additionally, the Office action contends that Lyle teaches providing information to the layout engine by which the layout engine preserves state information at column 4, lines 54-64 by disclosing an application control manager, and interrupting the layout engine, providing information to the layout engine by which the layout engine restores state from the state information, and restarting the layout engine from the restored state at column 10, lines 3-14 by again disclosing the transmitting of a prescribed function code to the application control manager to automatically initiate a process. Applicants respectfully disagree with each of these contentions.

As stated previously, the cited an applied portion of Lyle does not teach or event suggest using an engine, let alone using a highly-specialized type of engine

called a layout engine that is capable of laying out model elements. Further, the pushbutton process initiation is merely a user input interface. That is, the pushbutton is not capable of providing information about the state of the process. When activated, the system is simply interrupted without any regard to preserving information about the state of the process when interrupted. As such, not a single element of claim 19 is taught or even suggested by Lyle. For at least these reasons, applicants submit that claim 19 is allowable over the prior art of record.

With regard to claims 20-29, these claims depend either directly or indirectly from claim 19. Applicants further submit that claims 20-29 are also allowable for the additional patentable elements included in these claims.

For example, claim 28 recites that the status information includes data corresponding to time remaining to complete laying out the model elements. The Office action contends that Lyle fails to teach a system wherein the status information includes data corresponding to time remaining to complete laying out the model elements. The Office action goes on further to contend that Hurtado teaches a method wherein the status information includes data corresponding to time remaining to complete laying out the model elements. The Office action then concludes that it would have been obvious to an artisan at the time of the invention to include Wittenberg's teachings with Lyle's teachings in order to allow a user to use time more efficiently. Applicants respectfully disagree with this conclusion.

Applicants acknowledge that Hurtado may generally teach a time status indication. However, Lyle again does not teach virtually any element of claim 19, upon which claim 28 depends. That is, as shown above, Lyle does not teach using

a modeling engine, a layout engine, or an automatic layout process. Thus, the combination of Lyle and Wittenberg, even if somehow permissible, would not make obvious the recitations of claim 28. Applicants submit that claim 28 is allowable for at least this additional reason.

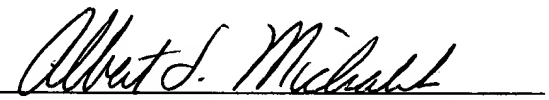
For at least these reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-29 are patentable over the prior art of record, and that the application is good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicant
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

CERTIFICATE OF MAILING

I hereby certify that this Amendment and Petition for Extension of Time, along with Transmittal, are being deposited with the United States Postal Service on the date shown below with sufficient postage as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231.

Date: March 17, 2004

Albert S. Michalik

2660 Amendment